

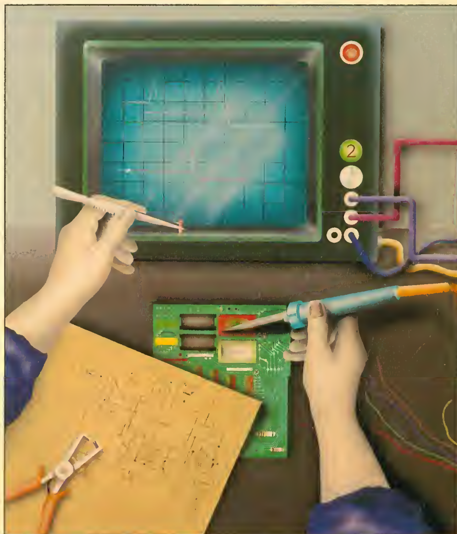
# SPECIAL

DM 24,–

4/81

Eine Sonderpublikation von CHIP. Das Mikrocomputer-Fachmagazin.

Österreich – öS 210,00; Luxemburg – lfr 430,00; Schweiz – sfr 26,00; Niederlande – hfl 30,00



# Bauanleitungen für Mikrocomputer

**Die besten Ideen, einen Computer auszubauen, zu erweitern und optimal zu nutzen.**

**Für AIM-65, Apple, CBM, TRS-80 und andere**

**Für alle bekannten Computersysteme**

Norbert Linder

# Schnelle multiplizierende und dividierende arithmetisch-logische Einheit

Ein bei allen Mikrocomputersystemen häufig auftretendes Problem ist die lange Rechenzeit für rein softwaremäßig realisierte arithmetische Operationen. Dies wirkt sich besonders bei Multiplikationen und Divisionen aus, die auch Voraussetzung für die Berechnung aller komplexeren Funktionen wie SIN(X) oder LG(X) sind. So liegt z.B. die Ausführungszeit für eine Gleitpunktmultiplikation im 32-bit-Format meistens schon im Millisekundenbereich. Bei umfangreicheren mathematischen Problemstellungen treten daher ziemlich lange Programmlaufzeiten auf, unter Umständen können manche Aufgaben wegen des hohen Zeitaufwandes gar nicht oder nur unbefriedigend gelöst werden. Ein Beispiel dafür ist die numerische Integration, die zur Erzielung einer ausreichenden Genauigkeit sehr viele Rechenschritte erfordert.

Eine Lösung des Problems ist die Verwendung von Arithmetikprozessoren. Von Vorteil ist, daß diese Bausteine neben den mathematischen Grundoperationen auch die wichtigsten komplexeren Funktionen ausführen können; als Nachteil steht der hohe Preis gegenüber.

Eine Alternative dazu kann die hier beschriebene Schaltung bilden. Sie läßt sich mit geringerem finanziellen Aufwand aufbauen, bietet aber nicht den Komfort eines Arithmetikprozessors.

Rein hardwaregesteuert kann sie sehr schnell einfache Multiplikationen und Divisionen ausführen; zusätzlich stellt sie alle Funktionen, die die verwendete ALU (74181) durchführen kann, zur Verfügung. Darunter fallen u.a. Addition und Subtraktion mit und ohne Übertrag, Inkrementieren, Dekrementieren und zahlreiche logische Verknüpfungen. Die Wortlänge der parallel verarbeiteten Daten kann praktisch frei gewählt werden, so daß die Einheit ohne weiteres für sämtliche Mantissenoperationen bei Gleitpunktzahlen beliebigen Formats verwendbar ist. Zusammen mit entsprechender Software können damit in einem Mikrocomputersystem alle Funktionen eines Arithmetikprozessors mit etwa gleicher Rechenzeit realisiert werden.

## Die Schaltungsprinzipien

Da die Gesamtschaltung relativ umfangreich ist, soll hier zunächst anhand vereinfachter Darstellungen die Funktionsweise der Schaltung bei den verschiedenen Operationsarten erklärt werden.

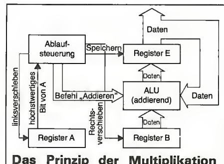
Man kann drei sinnvolle „Zustände" unterscheiden, die abhängig von den Werten der M- und D-Bit im Befehlswort sind. Je einer dieser Zustände ist der Multiplikation, der Division und den restlichen „einfachen" ALU-Operationen zugeordnet.

## Eine Multiplikation

Dem Rechenvorgang liegt das „normale" Multiplikationsverfahren für Dualzahlen zugrunde: Der Reihe nach werden alle Stellen des Operanden A abgefragt und abhängig von deren Wert die jeweils entsprechend verschobenen Darstellungen des Operanden B aufaddiert.

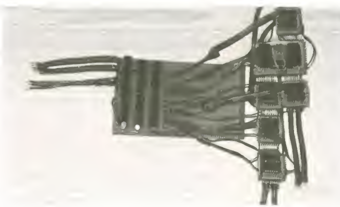
Dazu müssen der als Addierer arbeitenden ALU die parallelen Daten des Registers B und des Registers E zugeführt werden. Die Daten des Registers A gelangen nicht an die ALU; es wird von der Ablaufsteuerung nur das jeweils höchstwertige Bit 0 abgefragt.

Zu Beginn der Operation wird das Register E gelöscht. Der Addierer liefert an seinem Ausgang also den Wert des Operanden B. Wenn das Bit 0 von A auf 1-Signal 1 ist, veranlaßt die Ablaufsteuerung das Abspeichern des Additionsergebnisses im Register E; andernfalls findet dieser Vorgang nicht statt. Danach wird der Operand A um eine Stelle nach links, der Operand B um eine Stelle nach rechts verschoben. Dadurch rückt die zweite Stelle von A als höchstwertiges Bit nach. Am ALU-Ausgang steht jetzt die Summe aus dem vorher in E abgespeicherten Zwischenergebnis und dem verschobenen Operanden B. Das Additionsergebnis wird wieder abhängig vom neuen Bit 0 von A entweder abgespeichert oder nicht.

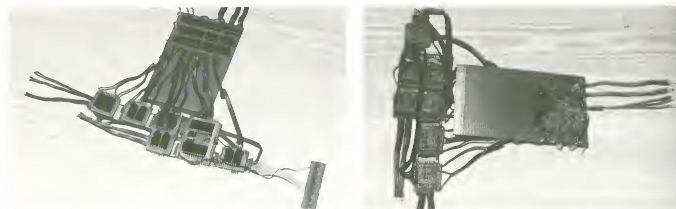



**Das Prinzip der Multiplikation**

Dieser Vorgang wiederholt sich so oft, wie die Anzahl der Stellen der Operanden beträgt. Überlegt man sich die hier dargestellte Arbeitsweise an einem praktischen Beispiel, so kann man erkennen, daß sie sich von den üblicherweise auf dem Papier durchgeführten Berechnungen nur insofern unterscheidet, daß bei jedem Rechenschritt eine Zwischensumme erzeugt wird, während man bei „manuellen" Multiplikationen die Zwischenprodukte am Schluß addiert.



**Schaltung der Arithmetikeinheit von oben. Auf der linken Seite die Ablaufsteuerung, rechts die als „Minimoduln" aufgebauten Register-, Multiplexer- und ALU-Einheiten.**



**Schaltung der Arithmetikeinheit im Testaufbau; Leuchtdioden am Ausgang** **Testschaltung der Arithmetikeinheit von unten**

## Die Division

Auch die Division erfolgt nach dem allgemein gebräuchlichen Prinzip: Operand B wird vom Operanden A abgezogen. Ist die Differenz positiv, wird sie als neuer Operand A übernommen, gleichzeitig wird an die Ergebnisdarstellung eine 1 angehängt. Falls die Differenz negativ ist, behält A seinen alten Wert bei, am Ergebnis wird eine 0 angetragen. Danach wird der gleiche Vorgang mit dem um eine Stelle nach rechts verschobenen Operanden B wieder ausgeführt.
Die Datenleitungen von A und B sind jetzt mit der subtrahierenden ALU verbunden (A – B). Die ALU-Ausgangsdaten werden an die Paralleleingänge des Registers A geführt.
Das Vorzeichen des Subtraktionsergebnisses kann aus dem Zustand des ALU-Carry-Bit abgeleitet werden. Daher wird es (invertiert) an den seriellen Eingang des Registers E und an die Ablaufsteuerung geleitet.
Nach dem Start der Operation erfolgt im E-Register ein Schiebevorgang nach links, wodurch das (invertierte) Carry-Bit in das Register übernommen wird. Außerdem speichert das Register A das Subtraktionsergebnis der ALU, falls der Wert des Carry-Bit anzeigt, daß es positiv ist. Anschließend wird der Inhalt des Registers B um eine Stelle nach rechts verschoben, und der Vorgang beginnt erneut. Er wiederholt sich wie bei der Multiplikation so oft, wie die Operanden Stellen haben.

## Eine „einfache" ALU-Operation

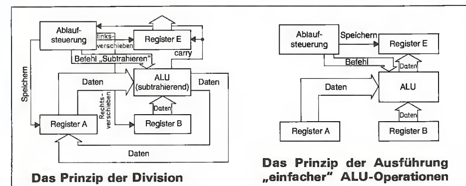
Bei der Ausführung der ALU-eigenen arithmetischen und logischen Funktionen sind die Datenleitungen beider Operanden an die entsprechenden ALU-Eingänge geführt, ebenso ist das Register E an die Ausgänge der ALU angeschlossen. Über das Befehlswort bestimmt die Ablaufsteuerung die Auswahl der ALU-Funktion. Bei der Operation wird lediglich das Register E am Anfang gelöscht und dann mit dem ALU-Ausgangssignal wieder beschrieben.

## Die Erweiterung

Die Schaltung ist für die Verarbeitung von Gleitpunktmantissen entwickelt worden. Dazu muß die Länge aller Register nur der Mantissenlänge entsprechen.
Es ist jedoch der höheren Rechengenauigkeit wegen sinnvoll, mit einigen weiteren Stellen zu rechnen. Deshalb sind alle Einheiten, die die schaltungsintern verlaufenden Paralleldaten verarbeiten, aufgeteilt in einen „Wortteil", der die Länge der Mantissen umfaßt, und in einen „Erweiterungsteil", der die zusätzlichen Stellen zur Verfügung stellt. Vom Mikrocomputer aus können bei den Registern A und B nur die Wortteile angesprochen werden. Für das Register E gilt ähnliches; es kann jedoch durch entsprechende Schaltungsauslegung auch eine Auslesung des gesamten Registers vorgesehen werden, wodurch die höhere Genauigkeit auch vom Mikrocomputer genutzt werden kann. Die Länge des Wortteils entspricht der Länge der Gleitpunktmantisse, die des Erweiterungsteils kann der Anwender selbst wählen.

## Die Gesamtschaltung

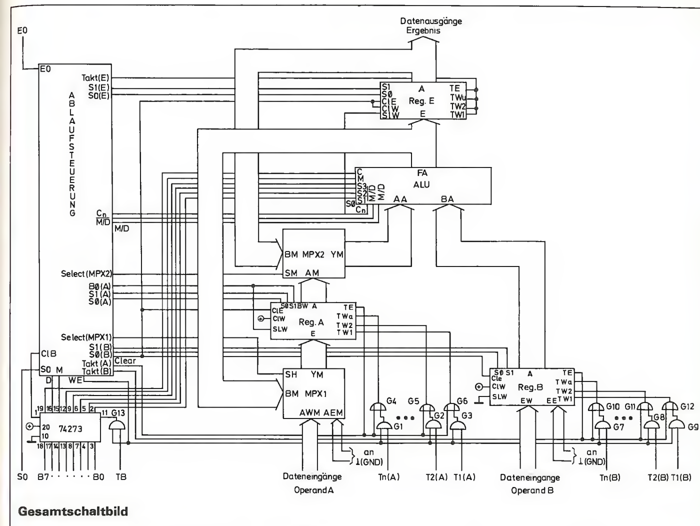
Die Gesamtschaltung zeigt den Aufbau der Elektronik. Der Übersicht halber sind die Register-, Multiplexer- und ALU-Einheiten sowie die Ablaufsteuerung als Blöcke dargestellt, deren detaillierte Innenschaltungen in separaten Bildern gezeigt werden. Außerdem wurden die parallelen Datenleitungen zwischen den Einheiten durch Pfeile ersetzt.
Zum Verständnis der Schaltung sollen zunächst die einzelnen Einheiten, mit Ausnahme der Ablaufsteuerung, besprochen werden. Sie verarbeiten alle die schaltungsintern parallel verlaufenden Daten und bestehen daher aus einem Wort- und einem Erweiterungsteil. In den Bildern ist die Trennung durch eine gestrichelte Linie kenntlich gemacht.
Da die Einheiten auf beliebige Länge ausgebaut werden können, wurde versucht, eine allgemeine Darstellung für sie zu finden. Dazu sind jeweils zwei Schaltkreise im Wort- und Erweiterungsteil in ihrer Zusammenschaltung gezeichnet. In die Lücke zwischen diesen und dem dritten können weitere Schaltkreise auf die gleiche Weise wie die beiden ersten angeschlossen werden. Der umrandete Teil kann in der Gesamtschaltung als „Black-Box" betrachtet werden; die Bezeichnungen der herausgeführten Ein- und Ausgänge entsprechen denen in den Detailschaltplänen.




**Das Prinzip der Division**

Ablauf-steuerung | Speichern | Register E | Befehl | Daten | Daten | ALU | Daten | Register A | Register B

**Das Prinzip der Ausführung „einfacher" ALU-Operationen**




**Gesamtschaltbild**

## Die Registereinheit

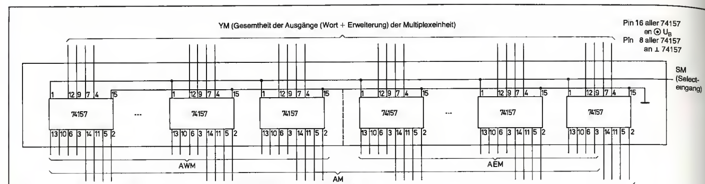
Alle drei benötigten Register A, B und E sind aus der gleichen Registereinheit aufgebaut.
Sie besteht aus 4-bit-Universalschieberegistern 74194, die, getrennt nach Wort- und Erweiterungsteil, so zusammengeschaltet sind, daß, neben dem Speichern natürlich, Rechtsverschieben des gesamten Registers und Linksverschieben der einzelnen Teile möglich ist. Dazu ist der „Shift Left Seriell Input" eines jeden Schieberegisters jeweils mit dem $Q_D$-Ausgang des vorherigen, der „Shift Right Seriell Input" mit dem $Q_A$-Ausgang des nachfolgenden Registers verbunden; an der Verbindungsstelle Wort-Erweiterung wird der Eingang zum Linksschieben als SLW(R) nach außen geführt.
Weiterhin sind extern zugänglich die Eingänge S0 und S1 (Auswahl des Arbeitsmodus: Speichern, Rechts-, Linksschieben), die Clear- und die Takteingänge.
Die Clear-Eingänge der einzelnen IC sind getrennt nach Wort-(ClW) und Erweiterungsteil (ClE) zusammengefaßt; bei den Takteingängen (TE, TW1, TW2 …TWn) kann zusätzlich jedes Schieberegister im Wortteil einzeln angesprochen werden. Am Ausgang B0(R) kann das höchstwertige Bit der Registereinheit abgegriffen werden.

## Die Multiplexereinheit

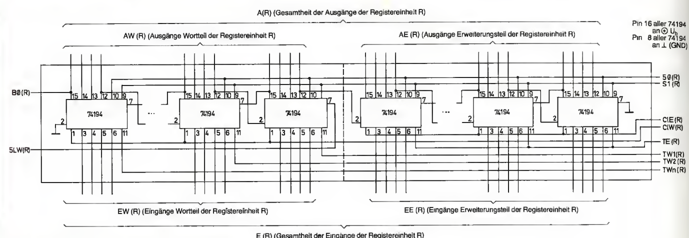
Für die Schaltung werden zwei identische Multiplexereinheiten benötigt. Es werden die TTL-Schaltkreise 74157 verwendet. Der Aufbau ist sehr einfach und direkt anhand des Schaltbildes verständlich: Die Daten von einem der Eingänge AM und BM werden je nach dem Pegel an SM (0 für AM, 1 für BM) an den Ausgang durchgeschaltet. Zwischen Wort- und Erweiterungsteil besteht kein Unterschied.

## Die ALU-Einheit

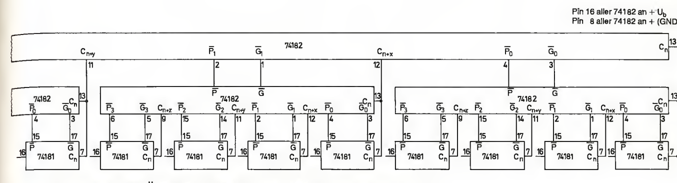
Die ALU-Einheit besteht aus den 4-bit-ALU-Bausteinen 74181, die parallel geschaltet werden.
Für die Übertragungsbildung gibt es zwei Möglichkeiten: Einmal die serielle Übertragungsbildung, bei der der Carry-Ausgang einer 74181 einfach mit dem Carry-Eingang der nächsthöheren verbunden wird. Diese Methode reicht für kleine Einheiten aus; ab einer Länge von 16 bit wird sie jedoch zu langsam. Dann empfiehlt sich die parallele Übertragungsbildung. Dazu wird zusätzlich die Übertragungseinheit 74182 benötigt, und zwar für Blöcke von vier ALU-Bausteine 74181 je eine, für vier dieser Blöcke wieder je eine usw. Zu beachten ist, daß die Übertragungsbildung für Wort- und Erweiterungsteil getrennt erfolgen muß.
An deren Verbindungsstelle übernimmt die gezeigte Logik aus den zwei UND- und ODER-Gatter die Übertragserzeugung für die erste ALU des Wortteils. Falls das $\overline{M/D}$-Signal auf 1 ist und damit anzeigt, daß weder eine Multiplikation noch eine Division erfolgen soll, wird das im Befehlswort enthaltene Bit für den Übertrag über den $C_0$-Eingang direkt an die erste Wort-ALU geschaltet. Im anderen Fall gelangt es nur an die erste ALU des Erweiterungsteils; zwischen Wort und Erweiterung wird der Carry seriell übertragen (M/D auf 1-Signal). Grund dafür ist, daß nur bei Multiplikation und Division der Erweiterungsteil mit benutzt wird.



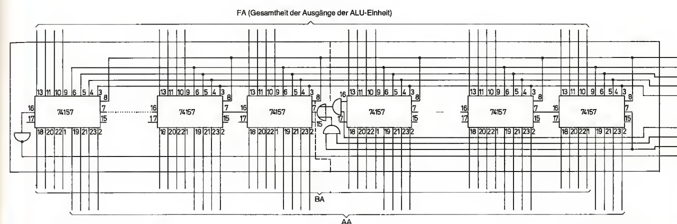

**Detailschaltung der Multiplexereinheit**




**Detailschaltung der Registereinheit**




**Prinzip der parallelen Übertragungsbildung (nach Unterlagen von TI)**




**Detailschaltung der ALU-Einheit**

Die Eingänge S0, S1, S2, S3 und M sind für die Funktionsauswahl verantwortlich.

$C_n$ ist der Übertragungsausgang der gesamten ALU-Einheit. Durch den Inverter wird er für die Verwendung bei der Division angepaßt ($C_n = 0$, falls Subtraktionsergebnis negativ, $C_n = 1$, wenn positiv).

## Der Aufbau der Gesamtschaltung

Die Gesamtschaltung besteht im wesentlichen aus den schon besprochenen Einheiten, der Ablaufsteuerung, dem Befehlsregister und der Logik zur Taktsteuerung der Register A und B, bestehend aus den Gattern G1 ··· G13.

Die Multiplexer MPX1 und MPX2 stellen die verschiedenen für die Multiplikation und die Division erforderlichen Anschaltungsweisen des Registers A an die übrigen Einheiten her.

In dem mit dem IC 74273 aufgebauten Register steht der 8-bit-Befehlskode für die gewünschte Operation. Befehl und Operanden können durch Taktimpulse an TB, T1(A) ··· Tn(A) und T1(B) ··· Tn(B) abgespeichert werden, solange keine Operation ausgeführt wird. Die Eingangsdaten für die Register A und B werden dabei an die Worteingänge gelegt (bei A an die des Multiplexers MPX1); die Eingänge der Erweiterungsteile sind mit Masse verbunden.

Eine positive Flanke am Eingang S0 bewirkt den Start des Operationsablaufs, das E0-Signal kennzeichnet das Ende.

## Der zeitliche Ablauf der Operationen

Nach jeder vollständigen Ausführung einer Rechenoperation werden alle Flipflops FF1 ··· FF4 in der Ablaufsteuerung und das Befehlsregister zurückgesetzt.

Damit sind die Eingänge S0 und S1 der Register A und B beide auf 1-Signal, die Einheiten also aufs Speichern vorbereitet. Außerdem führt die WE-(„Write Enable"-)Leitung einen H-Pegel. Der Multiplexer MPX1 schaltet wegen des L-Pegels am Select-Eingang die Daten vom Eingang A an den Ausgang durch. Vom Mikrocomputer können nun die Operanden und der Befehl in die Register geschrieben werden. Dazu müssen, während die Daten an den Eingängen liegen, Taktimpulse mit positiver Flanke an die Takteingänge T1(A)···Tn(A), T1(B)··· Tn(B) und TB gesendet werden. Dabei ist zu beachten, daß bei Multiplikation und Division das Befehlswort zuletzt abgespeichert wird, da mit dessen Übernahme der Multiplexer MPX1 keine Daten vom A-Eingang mehr weitergibt (bei der Division) und das Register B auf Schiebebetrieb umschaltet.

Eine positive Flanke an SØ startet nun den Rechenvorgang. Dabei wird FF1 gesetzt, das durch den Pegelwechsel an $\overline{Q}$ auch FF2 taktet. Während der beim Setzen von FF2 auftretenden Verzögerungszeit wird der Ausgang des ODER-Gatters N8 auf Ø-Signal und löscht damit die gesamte Registereinheit E und die Erweiterungsteile von A und B. Sobald $\overline{Q}$ von FF2 auf Ø-Signal geworden ist, wird FF1 wieder in seinen Ausgangszustand gebracht. Außerdem werden nun über die WE-Leitung (Ø-Signal) externe Taktimpulse gesperrt.

Der positive Pegel am Q-Ausgang von FF2 gibt das Flipflop FF3 frei. Mit jedem vom Taktgenerator, bestehend aus den Invertern N1, N2 und N3, kommenden Impuls werden jetzt dessen Ausgänge umgeschaltet. Dadurch entsteht ein symmetrischer Zweiphasentakt mit $Q = 1$ als erster und $\overline{Q} = 1$ als zweiter Taktphase.

FF4 liefert erst nach dem ersten Taktsignal einen H-Pegel an N18 und N22 und verhindert dadurch, daß bereits vor Operationsstart, während des Einschreibens des Befehls, über N15 oder N21 Taktimpulse an die Registereinheiten A und B gelangen können.

Der Q-Ausgang von FF3 wird an den Eingang des IC 74393 geführt. Die zwei in diesem IC enthaltenen 4-bit-Binärzähler sind zu einem 8-bit-Zähler zusammengeschaltet, der seinen Zählerstand mit jeder negativen Flanke am Eingang erhöht.

Eine „einfache" ALU-Operation wird innerhalb eines Taktzyklus ausgeführt. Da die Leitungen M und D beide L-Pegel führen, ist das durch N19 und N20 erzeugte $\overline{M/D}$-Signal auf 1-Signal. Bereits nach der ersten negativen Flanke am Zählereingang schalten daher N5 und N6 durch. Dadurch werden der Zähler und gleichzeitig über N7 das Befehlsregister, die Flipflops FF2 und FF4 und als Folge davon auch FF3 zurückgesetzt. Der Pegel des EØ-Ausgangs wechselt von L auf H und zeigt das Operationsende an. Dieser Zustand wird so lange aufrechterhalten, bis eine neue positive Flanke an FF1 eintrifft.

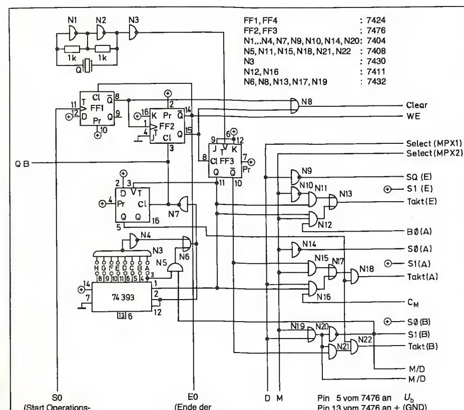
Bei einer Multiplikation oder Division soll die Reset-Routine erst beginnen, wenn der Zählerstand den Wert der Länge der Operanden erreicht hat. Dazu werden diejenigen Zählerausgänge, die diesen Wert binär darstellen, an das NAND-Gatter N3 geführt, alle anderen Gattereingänge werden auf H-Pegel gelegt (z.B.: Mantissenlänge 24 bit = 00011000, d.h. die Ausgänge D und E des Zählers an N3, alle anderen Anschlüsse von N3 an $+U_b$). Da N5 gesperrt ist, wird der Reset erst eingeleitet, wenn der Zähler den so vorgegebenen Stand erreicht. Innerhalb der eben beschriebenen Initialisierungs- und Reset-Routine laufen die eigentlichen Rechenoperationen ab. Die folgenden Erläuterungen zeigen die schaltungstechnische Realisierung der am Anfang theoretisch dargestellten Arbeitsweise.

## Der Multiplikationsablauf

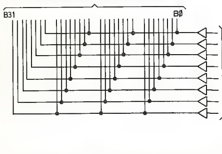
Bei der Multiplikation ist das M-Bit im Befehlswort gesetzt, das D-Bit auf Ø-Signal. Die ALU-Einheit addiert die Daten der Register B und E. Über den Multiplexer MPX2 wird diese am Anfang prinzipiell beschriebene Verschaltung der Datenwege hergestellt. Der Rechenvorgang läuft folgendermaßen ab: Über die S0- und S1-Eingänge ist das Register E aufs Speichern, A aufs Links- und B aufs Rechtsschieben vorbereitet.

Ist das höchstwertige Bit von A(B0(A)) auf 1-Signal, dann schalten während der ersten Taktphase die Gatter N12 und N13 durch. Da N11 gesperrt ist, erscheint eine positive Flanke am Takteingang der Registereinheit E. Diese übernimmt das Additionsergebnis vom Ausgang der ALU. Mit der zweiten Taktphase gelangen positive Impulse über die Gatter N15, N17, N18 und N19, N21 und N22 an alle Takteingänge von A und B und bewirken das Links- bzw. Rechtsverschieben der gesamten Registereinheiten.

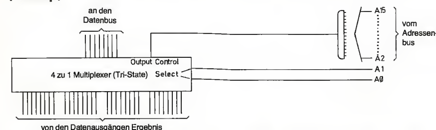
Dieser Vorgang wiederholt sich mit dem folgenden Taktzyklus so lange, bis der Reset eingeleitet wird.




**Schaltung der Ablaufsteuerung**




**Anschluß der Registereinheiten A und B an einen Mikrocomputer (Prinzip)**




**Anschluß der Registereinheit E an einen Mikrocomputer (Prinzip)**

Einen gewissen Vorteil bietet die Rückführung von B0(A) auf den SLW-Eingang: Der Operand A bleibt bei der Rechnung erhalten und kann sofort weiterverwendet werden.

## Der Divisionsablauf

Bei der Division ist das D-Bit auf 1-Signal, das M-Bit auf Ø-Signal. Dadurch werden die Ausgangsdaten der ALU über den Multiplexer MPX1 an die Eingänge der Registereinheit A gelegt. Das Register E ist auf Linksschieben umgeschaltet; es kann nur seriell den $C_n$ der subtrahierenden ALU-Einheit über den Eingang SLW übernehmen. Register A ist durch S0 und S1 aufs Speichern, Register B aufs Rechtsschieben vorbereitet.

Mit der ersten Taktphase wird die Einheit E über N10, N11 und N13 getaktet und übernimmt den Carry der ALU. Gleichzeitig schalten N16, N17 und N18 durch, falls der Carry auf 1-Signal, das Subtraktionsergebnis also positiv ist. Dadurch wird es im Register A abgespeichert. Dieser Vorgang entfällt bei negativer Differenz. Während der zweiten Taktphase wird nur noch das Register B nach rechts verschoben.

## Ein „einfacher" ALU-Operationsablauf

Weder das M-Bit noch das D-Bit im Befehlswort darf gesetzt sein, so daß die Registerdaten direkt an die Engänge der ALU gelangen. Außerdem sind alle Registereinheiten auf Speichern geschaltet. Während der einzigen Taktphase übernimmt das Register E, getaktet über N11 und N13, das an den Ausgängen der ALU stehende Verknüpfungsergebnis. Beide Operanden bleiben erhalten.

## Hinweise zum Nachbau der Schaltung

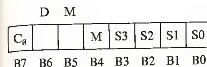
Die Schaltung ist vollständig in TTL-Technik ausgeführt. Man benötigt daher die übliche 5 V-Betriebsspannung. Dabei ziehen besonders größere Ausbaustufen relativ viel Strom. Das ist vor allem dann zu beachten, wenn das vorhandene Mikrocomputer-Netzteil zur Versorgung verwendet werden soll. Eventuell ist eine eigene Stromversorgungseinheit für die Schaltung vorzusehen.

Bei der praktischen Ausführung der Schaltung ist es empfehlenswert, die modulare Struktur beizubehalten und entweder Register-, Multiplexer- und ALU-Einheiten getrennt voneinander aufzubauen oder jeweils den die Daten verarbeitenden Teil für eine bestimmte Länge auf einer Platine unterzubringen. Letztere Methode hat den Vorteil, daß eine nachträgliche Erweiterung leicht möglich ist. Auf jeden Fall sollten die Ablaufsteuerung und das Befehlsregister auf eine eigene Platine gelötet werden.

Die maximale Rechengeschwindigkeit hängt von der verwendeten TTL-Technik ab. Meistens werden es LS-Schaltkreise sein; für die ALU und die Übertragseinheit ist es jedoch sinnvoll, reine Schottky-IC einzubauen. Auch die maximale Taktfrequenz wird damit hauptsächlich durch die TTL-Familie bestimmt, weniger durch die Größe der Einheiten. Bei LS-Schaltkreisen funktioniert die Schaltung noch einwandfrei bei 10 MHz, mit Schottky-ALU-Bausteinen sind auch 20 MHz möglich. Für besonders hohe Rechengeschwindigkeiten können die angegebenen Werte noch gesteigert werden. Dazu kann für Versuche der Quarz Q durch einen Kondensator ersetzt werden.



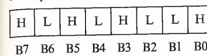




| $C_\emptyset$ | D | M | M | S3 | S2 | S1 | S0 |
|---|---|---|---|---|---|---|---|
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

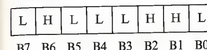
Aufbau des Befehlswortes

Multiplikation:



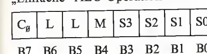
| H | L | H | L | H | L | L | H |
|---|---|---|---|---|---|---|---|
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

Division:



| L | H | L | L | L | H | H | L |
|---|---|---|---|---|---|---|---|
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

„Einfache" ALU-Operationen:



| $C_\emptyset$ | L | L | M | S3 | S2 | S1 | S0 |
|---|---|---|---|---|---|---|---|
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

### Befehlsworte für die verschiedenen Operationen

| | | | | M = H: logische Fkt. | M = L: Arithmetische Operationen | |
|---|---|---|---|---|---|---|
| | | | | $C_0$ = X | $C_0$ = H | $C_0$ = L |
| S3 | S2 | S1 | S0 | | | |
| L | L | L | L | F = $\overline{A}$ | F = A | F = A plus 1 |
| L | L | L | H | F = $\overline{A + B}$ | F = A + B | F = (A + B) plus 1 |
| L | L | H | L | F = $\overline{A}B$ | F = A + $\overline{B}$ | F = (A + $\overline{B}$) plus 1 |
| L | L | H | H | F = 0 | F = minus 1 | F = Null |
| L | H | L | L | F = $\overline{AB}$ | F = A plus $A\overline{B}$ | F = A plus $A\overline{B}$ plus 1 |
| L | H | L | H | F = $\overline{B}$ | F = (A + B) plus $A\overline{B}$ | F = (A + B) plus $A\overline{B}$ plus 1 |
| L | H | H | L | F = A ⊕ B | F = A minus B minus 1 | F = A minus B |
| L | H | H | H | F = $A\overline{B}$ | F = $A\overline{B}$ minus 1 | F = $A\overline{B}$ |
| H | L | L | L | F = $\overline{A}$ + B | F = A plus AB | F = A plus AB plus 1 |
| H | L | L | H | F = $\overline{A \oplus B}$ | F = A plus B | F = A plus B plus 1 |
| H | L | H | L | F = B | F = (A + $\overline{B}$) plus AB | F = (A + $\overline{B}$) plus AB plus 1 |
| H | L | H | H | F = AB | F = AB minus 1 | F = AB |
| H | H | L | L | F = 1 | F = A plus A | F = A plus A plus 1 |
| H | H | L | H | F = A + $\overline{B}$ | F = (A + B) plus A | F = (A + B) plus A plus 1 |
| H | H | H | L | F = A + B | F = (A + $\overline{B}$) plus A | F = (A + $\overline{B}$) plus A plus 1 |
| H | H | H | H | F = A | F = A minus 1 | F = A |

## Der Anschluß an den Mikrocomputer

Beim Anschließen der Schaltung an das Mikrocomputersystem gibt es prinzipiell zwei Möglichkeiten: Entweder verbindet man die Registereinheiten mit Ports und gibt über diese die Operanden und Befehle aus bzw. liest das Ergebnis ein, oder man baut sie als Speichereinheiten ein und spricht sie über den Daten- und Adressenbus direkt an.

Das Prinzip der zweiten Möglichkeit, die einfacher und bei der Verarbeitung schneller ist, soll hier kurz genauer erläutert werden. Als Beispiel dient der Fall, daß die Schaltung in einer 32-bit-Ausführung an einen Mikrocomputer mit 8-bit-Daten- und 16-bit-Adressenbus angeschlossen wird.

Die Verdrahtung für die Registereinheiten A und B ist einfach. Die Datenleitungen werden, vom Datenbus kommend, gepuffert und in vier parallel geführten Gruppen an die 32 Registereingänge gelegt. Die acht Wort-Takteingänge sind zu vier Paaren so zusammengeschaltet, daß bei einem Taktimpuls an eines der Paare entweder die ersten, die zweiten, die dritten oder die vierten 8 bit eingelesen werden. Ein 2-zu-4-Dekoder wählt aufgrund der 2 bit A0 und A1 des Adressenbus die entsprechende Taktleitung aus. Die anderen Adressenbit A2 ··· A15 werden in einem UND-Gatter so verknüpft, daß dieses nur bei einer bestimmten Konfiguration durchschaltet. Der Ausgang des UND-Gatters gibt mit einem H-Pegel den Dekoder frei. Dadurch können je acht bit des Registers genau unter einer Adresse angesprochen werden.

Die Auslesung des Registers E geschieht auf ähnliche Weise. Die Registerausgänge gelangen hier jedoch an die Eingänge eines acht bit breiten 4-zu-1-Multiplexers, der aus den vier Gruppen von je acht bit eine auf den Datenbus schaltet. Die Bit A0 und A1 bestimmen hier, welche der Gruppen ausgewählt wird, A2 ··· A15 werden, wiederum UND-verknüpft, an den Output-Control-Eingang geführt. Der Multiplexer muß unbedingt Tri-State-Ausgänge besitzen, die dann, wenn keine Auslesung erfolgt, hochohmigen Zustand annehmen.

Das Befehlsregister wird durch eine UND-Verknüpfung aller Adressenbit ebenfalls eindeutig adressiert.

Zu den Registern der Schaltung dürfen keine „normalen" Speichereinheiten parallel geschaltet sein; bei den Einheiten A und B wäre dies zwar theoretisch möglich, ist wegen der Gefahr eventuell auftretender Fehler nicht zu empfehlen.

## Hinweise zur Anwendung der Schaltung

Die Schaltung ist, wie schon mehrfach erwähnt, zur Verarbeitung von Gleitpunktzahlen gedacht. Es bestehen aber selbstverständlich auch weitere Anwendungsmöglichkeiten. Zu erwähnen ist hier besonders die Verwendung als n · n-Bit-Multiplizierer mit 2n-Bit-Ergebnis. Dazu müssen die Wort- und Erweiterungsteile beide gleich groß dimensioniert und beide ausgelesen werden. Damit läßt sich zusätzlich eine leistungsfähige Festkommaarithmetik realisieren.

Bei der Anwendung als Gleitpunktmultiplizierer oder -dividierer ist zu beachten, daß keine negativen Mantissen verarbeitet werden können. Vor der Operation müssen also eventuell einer oder beide Operanden negiert werden, was ebenfalls durch diese Schaltung geschehen kann. Bei der Division müssen sich außerdem die höchstwertigen gesetzten Bit der Operanden an der gleichen Stelle befinden, da sonst ein falsches Ergebnis geliefert wird. Dies muß gegebenenfalls vorher durch das Verschieben einer der Mantissen erreicht werden. Diese Verschiebung muß beim Ergebnis berücksichtigt und auf den Exponenten der Gleitpunktzahl umgerechnet werden.

Zum Schluß noch die Rechenzeiten: Eine 32-bit-Multiplikation/Division dauert mit einem 10-MHz-Takt ca. 6,4 µs, bei einem 20-MHz-Takt 3,2 µs.